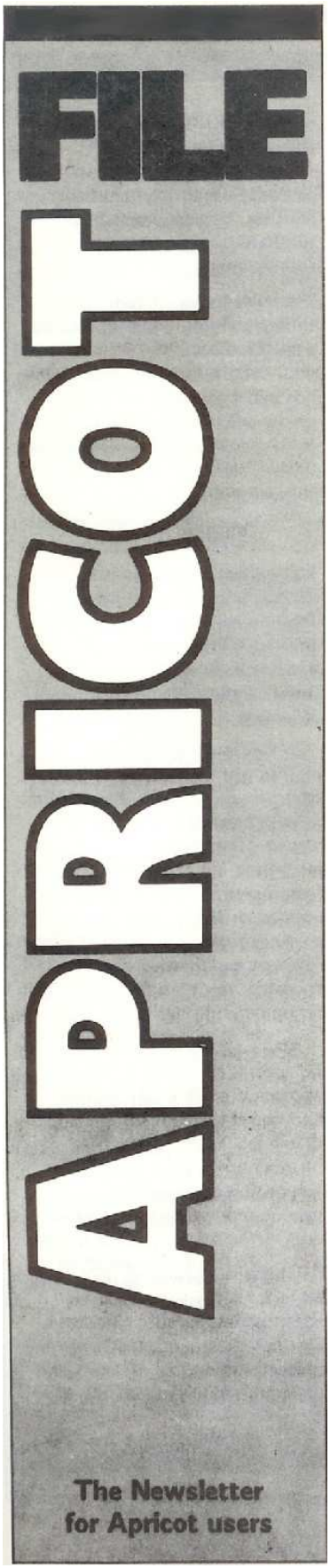
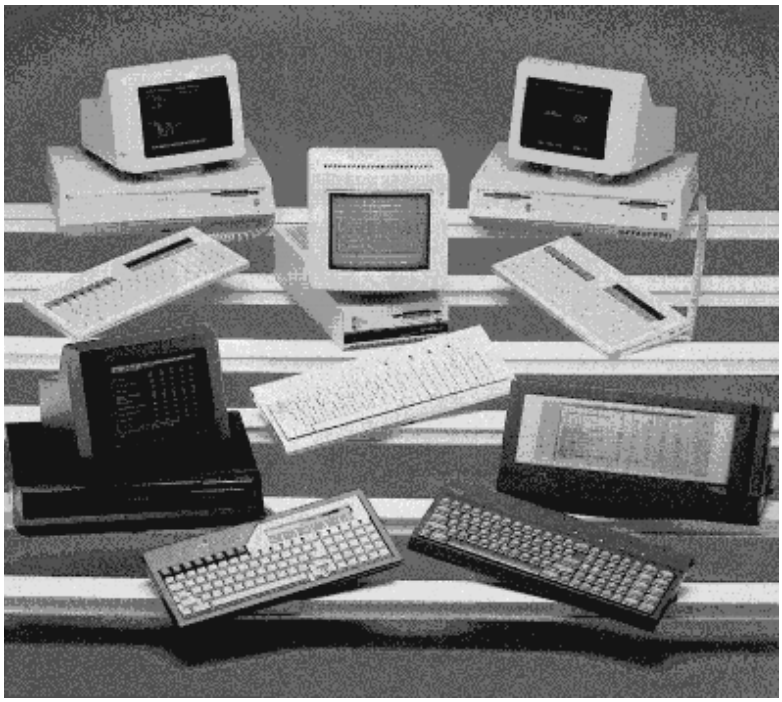


Apricot File 3.5



Scans courtesy of www.actapricot.org

Copyrighted content republished with thanks and permission from:

> Dennis Jarrett of TP Group
> Stuart Bunt
> Eddie Bromhead
> Mark Whitehorn

Many additional thanks to Eddie Bromhead for scanning all of Release 3, 4 and extra inserts.

Release 3.5

APRICOT FILE

The Newsletter for Apricot users

February 1988

Apricot junks Micromail

Artslink takes over e-mail operation

by Garret Keogh

Apricot Computers has handed over the running of its electronic mail service Micromail — which runs as a closed user area on Telecom Gold — to an outside company.

Apricot says it can no longer offer the dedicated support that Micromail demands; indeed, it does seem somewhat anomalous for a systems-oriented microcomputer company to be dabbling with such small-scale small-profit ventures. It makes much more sense for Apricot to retain a corner of Telecom Gold for its internal and dealer communications, and that in fact has been done.

Meanwhile, existing Micromail customers outside the Apricot ambit are being invited to resubscribe under the new regime. The company that has taken over Micromail, Artslink, has some experience; it was set up some two years ago to promote a Telecom Gold service within the performing arts world, but it obviously keen to spread its wings. Artslink describes itself as "one of the most experienced distributors of Telecom Gold, with a strong international base" and promises "to maintain and extend the services available to Micromail subscribers". It does have a substantial account base, which means more clout with Telecom Gold and (hopefully) better terms for subscribers.

John Beadle, Director of Artslink International, explains how he came up with the deal Apricot couldn't refuse. "Apricot wished to spend more time and money concentrating on shifting software and computers. We offered it a deal where we would take over the running of Micromail using our already dedicated base within the Telecom Gold world". Gail Capener at Apricot plc confirmed that Artslink's offer came at a time when Apricot felt it could no longer dedicate the time and energy to what is just another service.

The selling of Micromail affects something over 500 subscribers, in particular those with numbers starting MMU or MMD. Apricot's other series, numbers with a JET prefix, shouldn't be affected — with one or two exceptions, including possibly ourselves. JET was originally reserved for Apricot's own internal users, but a handful of outsiders (mostly journalists and periodicals, such as APRICOT FILE) also got involved. Apricot now wants to remove all of those JET users who are not strictly Apricot personnel; "JET is now essentially a company group".

Micromail users who do elect to join Artslink are promised a reduction in actual costs — "nearly all Micromail subscribers will benefit immediately by saving money" says Artslink. For a start, Artslink's Telex rates are around 20 per cent lower than Micromail's; and a monthly £10 standing charge replaces the £15 per month minimum. Extra Artslink services include text transmission from fax machine users as well as to them, a menu system that can bypass virtually all system prompts (good for beginners and temps), and a 24-hour seven-days-a-week Helpline. New services planned for this year include Interflora, Telemessages to North America, and on-line typesetting. ◆

In this issue

The Disinformation Column: On the Big Show 2
The Workshop: 132 columns on Apricots 6
Notebook: Alternative fonts retrospective 8
The Workshop: Customise your keyboard — Part One 9
The Data Protection Act and you 10
Turbo Pascal Primer — Part Four: Data types 16
Free Small Ads 20
Doodles by David 21
About the Newsletter 23

The Disinformation Column

On the Big Show

(and after)

By David Langford

Computer shows are grim affairs for the ordinary punter in the street. When I popped up to the Birmingham NEC for January's *Which Computer* jamboree, every hall seemed jammed with big outfits trying to flog big systems to big spenders.

One alleged theme was networking, that ingenious technological development which allows up to 32 expensive workstations to be linked together in simulation of one PC. Desktop publishing was the show's other theme, and I now estimate that the number of heavily hyped systems far exceeds the number of users who actually want to incorporate naff graphics into their multi-column, justified, mixed-typestyle, headline-spattered and footnote-ridden text, and then to buy the staggeringly expensive laser printer which will output the result with truly brain-bursting lack of speed.

Computer shows always leave me jaundiced, but *Which Computer* seemed grimmer than most. The bars and snacks weren't a patch on Olympia's (where the franchising system encourages a bit of variety in food and drink: I have fond memories of salt beef sandwiches at the last Olympia show, something I hadn't tasted since New York).

And, more to the point, the official information service was a marvel of computerised inefficiency. I went to ask where APRICOT FILE might be found. They sent me away with a form on which my desires had to be recorded for the System. The closest items discoverable on the form were 'magazines' and 'publications'. In scarcely ten minutes, the information desk's laser printer dutifully made me a neat little map showing all the magazine and publication stands I'd already read about in the glossy catalogue which had gone to press some months before ... but no others. You might have thought that the show's on-line database would be just the place to record late entries? You might have thought that since the show had (as it later emerged) given its own standard-format signboards to APRICOT FILE and other last-minute exhibitors or subtenants of shared stands, it would have made sense to key in a quick database entry for every added signboard? Ha bloody ha.

My next mistake was to ask at the several acres of Apricot plc display, where (not without compelling me to accept several pounds of literature on 386-chip IBM-compatible mouse-infested networked desktop publishing systems), a nice sales lady plumbed the depths of her Apricot-written BIOS and assured me that our favourite magazine was somewhere in Hall 4. This led to an hour-long, in-depth survey of the wonders of Hall 4, followed by a pause for some terrible beer and a further ransacking of the catalogue index for entries under 'Apricot File', 'Comms File', 'Olivetti File', 'Victor/Sirius File', 'Deals for Readers', 'Permapoint', 'TP Group', 'Dennis Jarrett', 'Roger Gann', 'PO Box 509' ... until after scanning in vain for 'Islington' I finally thought to look under 'L' for 'London Borough of Islington'. Whose stand was in the same hall as Apricot plc's. Which was Hall 5.

A real intellectual challenge, these computer shows.

After I'd found Roger and Terry plying their humble trade, the fun began. In between copious coffees: and with Roger gloating over his generic MS-DOS 3.2 disk (which would boot any Apricot except, curiously, the PC/Xi — but it didn't work on my F10, Roger): and while sensitive, artistic Terry was planning to escape the hurly-burly of computing via an illustrated epic walk retracing the footsteps of Alexander (or was it Napoleon? No, he never mentioned marching on either Moscow or his stomach): the plaints of many a lonely Apricot user were heard at the APRICOT FILE exhibition stand. Here are some of those I remember.

My MicroScreen's Bust, What Do I Do Until The Replacement Comes?

Software developers have largely treated the PC/Xi MicroScreen with justifiable distrust: the membrane keys are apt to go *spung* after a while, especially the most-used one on the left, and reliance on these keys would seem at first cringe to rule out an F series version of the program. Even Apricot's old ladder-menu programs recognised this by letting options be set via cursors as well as MicroScreen label selection. SuperWriter and WordPerfect duplicate their MicroScreen functions as CONTROL-letter combinations (SuperWriter) or single CONTROL-numberpad keystrokes (WordPerfect). We could effectively do the same for other programs by using one of the many available keyboard editors to duplicate the MicroScreen character codes (normally ASCII 185 to 190 inclusive) elsewhere in the keyboard. Practised typists welcome this since they generally prefer a real keystroke to a squidgy pressure pad, however enticingly the latter is labelled.

Of course this gets a mite more complicated when a specialist keyboard has already changed the MicroScreen. I have a little memory-resident utility which reports the actual characters sent to the keyboard buffer when you press a key, and this tells me that in WordPerfect (which I happen to be using right now) the first MicroScreen key produces ASCII character 182 — but character 183 if Shift is pressed as well. Ugh.

The real snag comes with the calculator. When it's on and the vulnerable left-hand MicroScreen key has given up the ghost, there's no obvious way to turn your calculator off again without

rebooting ... and the whole number pad, heavily used by some programs, is out of commission until then. Happily there *is* a BIOS (Control Device) function called 'Get/Set keyboard status' which points the way to an answer.

With that sublime thickwittedness we have learned to love in Apricot, 'Get status' will not answer the obvious keyboard questions covered by its IBM equivalent — whether Shift or CONTROL is currently being pressed. Instead it allows obscure manipulation of things like 'STOP mode' (haven't you always longed for a program that will press the STOP key for you?) and 'Calculator mode'. The syntax of this function call is a bit weird; and like several others sparsely described in Peter Rodwell's *Advanced User's Guide*, it was nervously omitted from the APRICOT FILE Release 2.10 list of 'all the Control Device calls in the latest version of the BIOS'.

(Oops! Checking this mid-1987 article by Nick Wilson I see it contains a major piece of misdirection: the information put in the SI register before a Control Device call is referred to as RET for 'Return data'. This should read DAT2 for 'Second chunk of data, where required.' The RET value is returned in AX.)

For those interested in assembler, and those who have read my Release 3.3 notes on memory-resident programs, here is how I bashed out the stopgap program ESCALC.COM. Its effect is to allow the calculator to be turned off with the ESC key if it's on, while the ESC key produces a normal escape character if the calculator is off. A highly compressed description follows.

ESCALC intercepts the dummy interrupt called when a key is pressed: Interrupt F9 hex. It checks that the AX register contains 011B hex (meaning that the Escape character is coming on its own — it would be 021B hex if this were the first character of the two-character sequences generated by the cursor keys, HOME, and the like). If so, it saves registers BX, CX, DX, SI and DI with PUSH commands, and makes a Control Device call to test the calculator status:

```
MOV BX,32h      ; the keyboard device ...
MOV CX,0Fh      ; 'get/set status' command ...
MOV DX,0FF00h   ; ask for status of
                  everything ...
INT FCh         ; and do the deed.
TEST AL,100b    ; examine Bit 2, which if set
                  means the calculator's on.
```

Note that although the bits to be tested are specified in the high byte of DX, the information comes back in the *low* byte of AX (a point made deeply unclear in the Rodwell book). Bit 0 represents something mildly obscure, Bit 1 the STOP key, Bit 2 the calculator; the rest are Shift Lock, Keyboard Translation (muck with this at your peril), Voice LED, Keyboard Lock (a good way to hang up your Apricot with official approval) and Caps Lock.

If we emerge from the Bit 2 test with the Z flag set, the calculator is off. A JZ jump takes us to a bit of program which restores the value 011B to AX and apologetically returns (after popping the saved registers). If the calculator is on, we turn it off with an almost identical function call, except that DX is loaded with FB00 hex instead of FF00. I think the logic here is that, since this time Bit 2 is not set in the upper byte, the zero in the lower byte becomes an active command to turn off the calculator. Anyway, it works. ESCALC then pops the saved registers, sets AX to zero to cancel the keystroke, and returns with IRET.

So much for the interrupt-handling part of this small, gem-like program. The installation part is routine — again, see the *Disinformation Column* of APRICOT FILE Release 3.3.

Turbo Pascal fans can achieve the same effect within their own programs by setting up the 'Regs' register record suggested in the manual, and using the built-in procedure call *Intr($FC,Regs);*. But I wouldn't attempt to write memory-resident code in Turbo. Lazy sods with duff MicroScreens may prefer to send me a small bribe, or a large one.

How Can I Print In Cyrillic?

I'd come across variants of this one before, and with the intense sloth so characteristic of Langford had invariably fled screaming. Now, from the evil way Roger had said, "Oh, Dave Langford will be able to do that for you easily, har har," I knew my reputation was on the line.

It sounds a boring but straightforward task, assuming an Epson-compatible matrix printer. The Russophile in question owned just this, and was actually using a cheap-and-nasty version of what I imagined — a program which printed out files in special fonts, but which had to be run separately from your word processor, meaning you had to save in plain ASCII format.

Obviously the next step was to write a memory-resident program which ...

(a) intercepted the BIOS calls via which MS-DOS prints characters;

(b) stored the current line of print in a temporary buffer until the line-end was signalled by Return;

(c) passed line feeds through to the printer unscathed;

(d) blocked printer escape sequences and took its own appropriate actions;

(e) examined the BIOS font address pointer to locate the font now in use;

(f) set an appropriate printer graphics mode from the several available on standard Epsons;

(g) translated font graphics into printer graphics for each accumulated characters, and sent the result to the hapless printer;

(h) reset everything and started again at (b).

In the week following the Show, I got a working model of this program (titled, with massive originality, FONTPRN) operational at Ansible Information HQ. It functioned surprisingly well, although item (d) above wasn't fully implemented just yet (because that would be *Work*). For the PC/Xi/Xen I had to set a vertical pitch eight printer dots high, and make two passes at (g) to print the top and bottom halves of the tall 10x16-dot fonts. This had the happy side-effect of making the Apricot 'line graphics' characters (ASCII codes 176 to 223) join up perfectly on the printout, but set the vertical pitch at a spacious five lines per inch.

Horizontal pitch could be changed quite a lot by swapping printer modes using the Epson's ESC * *n* sequence, where the mode *n* is a character in the range 0 to 4, or 6: the resulting pitches are 6, 12, 12*, 24*, 8 and 9 cpi — with the starred ones a bit dodgy since some dots are lost (horizontally adjacent dots can't be printed in these modes). Every character of the font, even the peculiar ones below 32, would print out — except characters 10, 13 and 27, because FONTPRN reacts specially to Linefeed, Return and Escape.

The F series version was simpler in several ways: with an 8x8 font you need only one pass per line and can leave the vertical pitch unchanged unless, again, you particularly want the line graphics to join up. One minor frustration was that the bits in the F-series font are stored 'the other way round' from those in PC/Xi fonts — my first efforts produced some marvelous mirror-printing, just like *Jabberwocky*. The 8x8 fonts also dictate a whole new series of horizontal pitches: corresponding to the six mentioned above, you now get 7.5, 15, 15*, 30* (a sight for sore eyes!), 10 and 12.5 respectively. But enough of this boasting.

The only snag is that the chap who wanted Cyrillic print made one slightly alarming restriction. The method employed, he laid down, must not use graphics; graphics have the fatal defects of (a) being too slow, and (b) looking like graphics. Trust me to encounter someone who wants to print graphics without using graphics. I dunno: I'm not too unhappy by the speed of the draft FONTPRN, and its PC/Xi font copying produces a quite nifty NLQ look in double-density graphics mode (on the 8x8 F series the result looks strangely like ordinary matrix print), but suspect the no-graphics restriction means the only way out is something like an Epson LQ1500 with downloadable character fonts — all too probably ruled out on the ground of being (c) too expensive. I'd be interested in some input on all this.

How Do I Make My Turbo Programs 'Configurable'?

The idea involved here wasn't a matter of allowing the configuring of compiled programs to different machines (Borland's Turbo Toolbox provides a way of doing this, but as far as I can see it rules out any fancy Apricot screen control work — better to compile a separate version for IBMs) but of setting variables 'semi-permanently' without recompiling all the time.

For example, your home-grown word processor might have a whole menu screen to enter printer information — what escape sequences turn on boldface, underlining, superscript, and so on: and likewise what turns them off again. Obviously the information would normally be all lost when you leave the program.

The standard solution would be to write file input and output procedures which duly saved the required data and loaded it again. But then you've got another data file knocking around the disk, and there's an extra delay while it loads up each time, and the whole thing seems a mite cumbersome when you mostly use the same printer anyway.

When asked, I came up with a simple Turbo alternative which (ultimate naughtiness!) allows the .COM file to modify its own internal data — the preset variables or, as Turbo confusingly calls them, typed constants. With a bit of luck, the attached program listing (placed I don't yet know where, in accordance with the editorial Design Standards and sense of Aesthetic Thingy) will be fairly self-explanatory.

INSTAL.PAS is clearly only a demonstration program, a housing for the procedure *InstalTypedConstant* — the only procedure featured, so it shouldn't be too hard to find. It expects the main program to have defined a string called *ProgName* which contains the MS-DOS name of the compiled program, here INSTAL.COM.

A call to *InstalTypedConstant* has two parameters. One is a variable, which *must* be a code segment variable — meaning it must be defined as a 'typed constant' rather than with *Var*. The other is the number of bytes you want copied from the current data back to the .COM file. For example, to instal a single code segment variable of any type or size whatever, this call will always work:

```
InstalTypedConstant(VariableName,
SizeOf(VariableName));
```

... as in the sample program itself. More likely you'll want to save several variables back to the .COM file at the same time. In this case, put them all together in succession under the *Const* heading. Then give *InstalTypedConstant* the name of the first variable, and a byte-count which is the total size of all the variables involved. The calculation of this can be automated, too: if the variables which are to be saved are *FirstVar*, *SecondVar*, *ThirdVar* and so on, and the first subsequent variable which *isn't* to be saved is *UnwantedVar*, then the number of bytes to save will be

```
Ofs(UnwantedVar) — Ofs(FirstVar)
```

I've arbitrarily limited the number of bytes installable to 1,000, the value given to

MaxInstalBytes. Increase this constant if you like, and if you have the patience to enter programs whose typed constant sections predefine more than 1,000 bytes of data ...

All the wretched INSTAL program does is allow you to set different bytes of the *Options* array to random numbers by pressing the number key with the appropriate array index. Pressing Escape will leave the program with its .COM file unchanged; pressing 'I' for Install will write the current values back to the .COM file, to be taken as initial values next time the program is run. Note that *InstalTypedConstant* will go beep if it can't find the .COM file it wishes to pervert — a likely occurrence if you test this by running it in memory without compiling it to INSTAL.COM.

Read the following precautions before you rush ahead!

(a) If you have another program called INSTAL.COM lying around, call this one something else — change *ProgName* and the MS-DOS filename (no need to change the name at the head of this Pascal source code). Otherwise horrid miscegenations might take place.

(b) If you install too few bytes, some of your settings will understandably be lost. Install too many, and weird things may happen — most likely, the changing of the initial setting of some variable(s) you *didn't* want fiddled with.

I am now going on holiday to the secret flat in North Wales where my wife forbids computers. Farewell, cruel software world ... ◆

```
Program INSTAL;

{Demonstrate Turbo Pascal .COM file 'self-installation' technique.}

Const
 Esc = ^[;
 Options: Array[1..6] of Byte = (1,2,3,4,5,6);
 ProgName: String[13] = 'INSTAL.COM';
Var
 Key: Char;ndex: Integer;

Procedure InstalTypedConstant(Var InstalVar; BytesToInstal:Integer);
Const
  Beep = #7;
  MaxInstalBytes = 1000;
Var
  InstalPointer: Integer;
  InstalFile: File of Byte;
  DummyArray: Array[1..MaxInstalBytes] of Byte absolute InstalVar;

begin
  Assign(InstalFile,ProgName);
  {$I-}
  Reset(InstalFile);
  {$I+}
  If IOresult>0 then begin
  Write(Beep);
  Exit
end;
If BytesToInstal>MaxInstalBytes then BytesToInstal:= MaxInstalBytes;
Seek(InstalFile, Ofs(InstalVar)-$100);
For InstalPointer:= 1 to BytesToInstal do
  Write(InstalFile,DummyArray[InstalPointer]);
  Close(InstalFile)
end;

Begin
  ClrScr;
  Repeat
  GoToXY(1,1);
  For Index:= 1 to 6 do
    Writeln('Option #',Index,'      ',Options[Index],'   ');
    Write('Select: ');
    Read(Kbd,Key);
    Key:= UpCase(Key);
     Case Key of
     '0'..'9': Options[Ord(Key)-48]:= Random(256);
     'I':      InstalTypedConstant(Options,SizeOf(Options));
    end;
  until Key in ['I',Esc];
End.
```

132 columns on the Apricot screen

Revisiting the Parthenon (columns, geddit?)

By Eddie Bromhead

One of the endearing features of the Sirius was just how few disks came with it. One contained CP/M-86, the other MS-DOS. Given that MS-DOS was version 1.25 and therefore had fewer utility programs (in the form of external commands), and that a single-sided Sirius floppy held 600K, this was easy enough. What's more, the Sirius had virtually no bundled software — no Super series, no GSX: everything was an extra.

There was one superb freebie on the disk, however. It was a program for switching the Sirius' high-resolution display into a 132-column mode — 132 characters per line, 40 lines per screen.

For a number of years I thought that this had no analogue on the Apricot. But eventually I found the very same utility on a Black Hole disk, and in Apricot format too (porting the Sirius program to the Apricot just didn't work). It makes sense; the Apricot is blessed with the same 800x400 screen as its illustrious predecessor, although it is configured very slightly differently. The differences are not so great that any Assembler programmer couldn't make the changes in minutes, but they're big enough to stop you transferring some software — especially when you don't have the source code.

The Black Hole

Black Hole disks were available to (and from) Apricot dealers, and contained a number of useful utilities which although released by Apricot were not officially supported by the company. They are often referred to in the pages of APRICOT FILE, and much Black Hole software has found its way into the DEALS FOR READERS lists. Inevitably, there is much of no relevance to an individual reader: but amongst the dross there is something for everyone.

The 132 column utility comes as two .ASM files; one contains the code, the other has the bit-image patterns of the font. These are 6x10 fonts (10x16 fonts are normally used on the PC/Xi screen). You only need assemble the 132C file, as the character set is INCLUDEd in this. Then you LINK the resulting .OBJ file (ignoring the 'no stack segment' message) and convert it from 132C.EXE file to a .COM file.

The source code contains within the Copyright Notice a credit to Nick Wilson, who is an occasional contributor to APRICOT FILE. From a completely different source, I obtained a text file on using 132C by John Bunney *(also an erstwhile contributor — Ed)*. This lists the Escape sequences which 132C supports. Most of the common ones are there, but some obviously have no relevance to the 132C mode — you can't swap fonts, for instance.

Assembly and the rest

To assemble and link (without generating listings) you need the commands ...

```
MACRO86 132C,,NUL;
LINK 132C,,NUL;
```

(Microsoft's Assembler is normally named MASM; mine was supplied with a different name for some strange reason. 'MACRO86 is the trading style of ...'.)

The conversion ought to be done with the utility EXE2BIN — although I've never found that it works. Instead, use DEBUG. This is the form:

```
A>  DEBUG 132C.EXE
    — N 132C.COM
    — W
      WRITING nnnn BYTES
    — Q
A>
```

Step 1 is invoking DEBUG, which loads the EXE file and fixes up all the relative addresses. Rename the file in memory with the N-for-Name command and Write it to disk, then Quit. (For anyone else who does EXE to COM conversions using DEBUG, I've found that it only works if your Assembly code has an ORG 0100H instruction at the start of the code.)

132C.COM is a terminate-and-stay-resident program which uses up about 7600 bytes. Unlike other TSR programs which are invoked and cancelled by 'hot keys', 132C is deactivated and reactivated by two small COM programs ON and OFF which are also supplied on the Black Hole disk. Basically, they each send an Escape sequence to the screen driver. One switches the 132-column utility off; the other, which only works when 132C is loaded, switches it back on. You can invoke these functions from within your own programs via Escape sequences.

When first loaded, 132C switches the screen into 132 column mode. Normality is restored with OFF, and back to compression with ON. And that, you might think, is that!

Using 132C

A number of programs will work in this special mode, some requiring reconfiguration; and some programs will not work with 132 columns at all.

In the first category are programs which let the Apricot take care of line-wrapping — Fortran and Basic programs being typical offenders. For

example, suppose you have a program which sends 130 characters per line to a wide carriage printer. With 132C loaded, you can see the formatting of this on-screen, and so save a lot of paper in the development stage.

In just the same way as you might switch from 10cpi to compressed mode on a printer by means of an Escape sequence, you can do the same with 132C once it has been installed. The Escape sequence corresponding to OFF is Esc z (conveniently the same as setting screen to its power-on state in normal mode); and the equivalent for ON is Esc | (the bar character is at CTRL-0 on the standard Apricot keyboard; the convenient engraving on the front of the key has been left off Xen keyboards, along with the tilde mark — it still works on the default keyboard, though).

In the second group — software which requires reconfiguration — you'll find programs like WordStar and SuperCalc. WordStar, for instance, can be installed to make use of the extra screen space: to work properly, that installation *must* be done — although WordStar will pack itself nicely into the top left of the display, it doesn't know that the status line has been whipped off to line 40 and some of the displays don't clear properly. Do the reinstallation using INSTALL or WINSTALL (depending on version, and see my comments below).

Programs which definitely don't like 132C are those that use alternate character sets (try the ladder-based programs from the MANAGER suite!) and those that use graphics.

Now, 132C does a number of things:

- First, it strips the high bit from every byte. This 'unWordStarring' saved its author from devising bit patterns for characters 128-255 in the ASCII set.
- Second, the symbols for the normally unprintable characters at the low end of the set are different from those on standard fonts.
- Third, the Escape handling is limited to VT52 escape codes — and those are a subset of the set which the PC/Xi has.
- Fourth, the MicroScreen echo is handled through a BIOS interrupt rather than via the Apricot Control Device.

The net effect of these are that you can't get special characters: or, if you put the screen into 'Literal' mode, they won't be what you expect. The lack of ANSI escape-handling isn't a problem normally, except that CLS in a batch file won't work any more. It's the same as loading the IBM emulator; for that matter, 132C and the emulator are incompatible.

The MicroScreen problem means that MicroScreen labelling doesn't work — certainly not for programs generated using RINSTALL. Every bit of documentation I have implies that you should use the Control Device rather than BIOS interrupts. This section needs to be rewritten for the best effect. On my PC the MicroScreen labels appear and the LEDs come on; but the information scrolls rapidly off the screen, and the date/time display is restored.

Screen refresh is slow on a Xen; on a PC, you really have to be *very* patient — I haven't timed it, but the John Bunney notes quote eight times slower. The characters are pretty small, even on the 12in monitor. On a 9in screen, they need a magnifying glass, but they are *very* sharp!

Xen problems

The 132C program wouldn't work properly on my Xen. It seemed to load, but the screen would be subject to such a rapid flicker when in compressed mode that it became unreadable. To fix this took some time, and some reading of the *Xen Technical Reference Manual* (on DEALS FOR READERS disk AP034).

The Xen has two screen modes: a native Xen mode, and a PC/Xi emulation mode. The main difference is that the video RAM and pointers are kept in different places, and are accessed through different ports. You have three choices: switch to PC/Xi mode using another program, add the code to 132C to switch the Xen into PC/Xi mode, or modify 132C to work in Xen mode.

I chose the latter. It's simple enough to do: merely alter the single occurrence of OUT 68H to OUT 88H. That sends the Cathode Ray Tube Controller (CRTC) chip its configuration data in the correct way. (Incidentally, the data Nick Wilson chooses to send isn't the same as is suggested in either the PC or *Technical Reference* manuals; but since their suggestions were no better — and no worse either — I left it very much alone.) Immediately, 132C worked on the XEN.

Xen monochrome screen

The Xen was designed to work with three graphics modes — PC/Xi standard monochrome, F-Series 640x200 colour, and 640x350 colour (the output from which is EGA-compatible, although the video board is much better). A bit more top-end RAM was pinched for pointers. That's why you can stretch the RAM on a PC to 960K (though nobody normally goes beyond 896K), but the Xen is limited to 896K.

Three ports were included in the video hardware, termed B, C and X. The addresses of B and C change in Xen or PC/Xi modes, but port X is always at 6CH. The two least significant bits of a byte sent to this port control whether the display is Xen or PC/Xi and how inverse video is handled; basically, this enables you to reverse the video screen permanently.

By using this in small COM programs, it is possible to get one of four permutations:

- OUT 6CH,00B Xen mode, normal video
- OUT 6CH,10B Xen mode, reverse video
- OUT 6CH,11B PC/Xi mode, reverse video
- OUT 6CH,01B PC/Xi mode, normal video

So four tiny COM programs could be devised. My experience is that setting PC/Xi mode causes the Xen to hang up, so I can only recommend the two Xen mode commands.

```
TITLE   Program to invert video display on XEN
Cgroup  GROUP   Code
Code    SEGMENT public 'Code'
        ASSUME  CS:Cgroup
ORG 100H
MOV AL, 010B      ; the corresponding
OUT 6CH, AL       ; normalise command
MOV AH,4CH        ; requires 000B
INT 21H
Code    ENDS
        END
```

Although the inversion is supposed to work in graphics, loading GEM, WINDOWS or even GSX will cancel its effect. Also, loading 132C will reset the screen behaviour to normal (although a subsequent invocation of the inversion program will take effect. Video reversal will only work if your graphics application performs this operation.

Footnote on installing WordStar

I have WordStar 3.30. On a machine with more than 512K, the installation program refuses to run — too little memory, it says. I got round this the first time by using Concurrent CP/M-86. That said, I had zero RAM; but at least the WordStar install program ran (WordStar itself didn't). Later, I got smarter and temporarily used a RAM disk to reduce available memory below 512K. Still later, I found the patch for WINSTALL.COM on WordStar 3.30b in the *Dealer Manual*:

```
DEBUG WINSTALL.COM
  - A2D4
  XXXX:02D4 JB 2DE
  XXXX:02D6
  - W
    WRITING XXXX BYTES
  - Q
```

Fat lot of good it did me, I seem to have 3.30a! ◆

For anyone who remembers 132C because they still have a Sirius, TP Group produces a Deals For Readers mail-order catalogue specifically for users of the Victor 9000, Sirius 1 and Vicki. Drop us a line for your free copy.

The Notebook

Alternative Fonts:

A Retrospective

By Eddie Bromhead

APRICOT FILE Release 2.6 ran my article on the subject of loading and using alternate fonts 2 and 3 on the Apricot PC/Xi under BIOS versions prior to 3.1. Subsequently I had some difficulty reconciling what I wrote with the Escape Codes series by David Langford. Basically, I hadn't appreciated the complete story; my table of escape calls needs amplification in respect of Esc *.

The sequence Esc * 0 will switch back to font 1, the same as SI. Similarly, Esc * 1 will change to font 2; and Esc * 2 will change the display to font 3.

The confusion arises because Esc * followed by a string also has a meaning. If the first character of the string is a space, the escape sequence assumes the same as Esc * 2 — it changes the font to no. 3 and doesn't print the space. Where the sequence Esc * is given on a line of its own, say in an Basic direct command, it also acts as if Esc * 2 had been given.

I've tried out all the combinations of Esc * and my original table should be expanded as follows:

Sequence	ASCII numbers (decimal)	Purpose
SI	15	Font 1 (default)
Esc * 0	27 42 48	Font 1 (default)
SO	14	Font 2 (first extra)
Esc * 1	27 42 49	Font 2 (first extra)
Esc * 2	27 42 50	Font 3 (second extra)
Esc * n	27 42 n	where character <n> has ASCII code 52 or greater, the character is ignored and font no. 1 is selected except if n=48 or n=49 (see above), where character <n> has ASCII code 51 or lower the character is ignored and font no. 3 is selected. If <n> is TAB, font no. 3 is selected and a TAB is performed as well

It should be obvious, too, from my descriptions that my early-model Apricot PC had version 1 of FONT and KEYS. Version 2 of these utilities is invoked in a somewhat different manner: you give the name of the font or keyboard file both as input to the utility, and also as a location where the currently installed font or keyboard might be saved.

For example, under version 1 the command FONT NOVEL loads NOVEL.CHR and saves the current font. FONT RESTORE reverts to the saved font. With version 2, the equivalent command is FONT NOVEL OLD — this loads NOVEL.CHR and saves the existing font as OLD.CHR. ◆

Customise your keyboard

Part One: MS-DOS command line editing

By Guy Sturges

There's no place like home. Have you noticed that when using someone else's terminal or PC that things aren't as you expect them to be? Keys are in different places and do different things. You miss that feature you added on your own machine to the CONTROL-CLEAR key which does exactly what you need now — and have to spend time working out how to do it on this strange machine.

I use a lot of different machines, and occasionally I find new functions that would be useful on my machine. As a result I have accumulated many useful extras for my Apricot.

The first function that I found missing on the Apricot was the MS-DOS command line editor. Admittedly there are some sophisticated memory-resident routines that allow you to recall previous commands; but I wanted simply to be able to repeat the last command possibly making a correction, if there was a mistake. This is really quite simple and I am surprised that Apricot didn't provide it as standard.

In the *Microsoft Pack* manual that comes with the machine there is a description of the awful EDLIN (though I have seen worse editors). In this you will find a page on 'special editing keys'. As its name implies, ED(it) LIN(e) edits a line, or 'template'. MS-DOS commands are stored in a template when the Return key is pressed before they are executed. The miraculous thing is that the previous command or template can be edited as though it was a line in EDLIN.

The special keys basically work by transferring characters from the template to the current cursor position in the line or skipping over characters in the template. There are two pointers: one is the cursor in the line being edited, the other is a pointer to a character in the template.

Without the manual at hand you'll never remember the codes you need, so to implement this effectively it's best to assign these sequences to logical keys on the keyboard utilising the ACTIVITY keyboard editor or KEYEDIT. I use the following:

CONTROL with CURSOR RIGHT
[equivalent to ESC S]
copy the character pointed to in the template into the current cursor position

CONTROL with CURSOR DOWN
[equivalent to ESC U]
copy all the remaining characters from the template pointer position to the end of the template to the cursor position

CONTROL with DELETE
[equivalent to ESC V] advance the template pointer right one position — ie skip over a character in the template

CONTROL with INSERT
[equivalent to ESC P]
toggle command line insert mode — ie add characters to command line without moving template pointer

CONTROL with CURSOR UP
[equivalent to ESC T]
copy all characters from the template pointer up to but not including the character specified after Escape T

CONTROL with CURSOR LEFT
[equivalent to ESC W]
advance the template pointer up to the character specified after Escape W

The template pointer and cursor positions move accordingly as the commands are executed.

So to repeat the last MS-DOS command, press CONTROL-cursor down and the last command will be entered at the command prompt. Suppose the last command was ...

```
SYMDEB MYPROG.EXE PARAM1 LAST PARAM
```

... and you wish to change it to ...

```
SYMDEB YOURPROG.EXE PARAM2 PARAM3
```

... the following sequence would do it:

- CONTROL-CURSOR RIGHT seven times — puts the first seven characters on the screen, produces SYMDEB <space>
- CONTROL-DELETE twice — skips two characters of the original
- CONTROL-INSERT YOUR — to add characters 'YOUR'; screen now shows SYMDEB YOUR
- CONTROL-CURSOR UP 1 — to copy original characters up to '1'; screen now shows SYMDEB YOURPROG.EXE PARAM
- CONTROL-DELETE — to skip next character
- CONTROL-INSERT 2 — to insert '2'; screen now shows SYMDEB YOURPROG.EXE PARAM2
- CONTROL-DELETE five times — to skip next five characters
- CONTROL-CURSOR DOWN — to copy all remaining characters; screen now shows SYMDEB YOURPROG.EXE PARAM2 PARAM
- type 3 — screen now shows SYMDEB YOURPROG.EXE PARAM2 PARAM3
- press Return to enter this line and execute the command.

This is just an example of the different features available, though this particular usage would be somewhat cumbersome in real life. Using just one feature to correct a misspelt name or some other error is really much more useful.

Try it first using the Escape sequences (remember that upper-case characters follow the ESC). When you are satisfied that it is useful, modify your keyboard more permanently with ACTIVITY or KEYEDIT.

Next month we'll be looking at customising the keyboard to handle the extended European character set with 'dead' key sequences and a multilingual keyboard driver. ◆

Processing the Personal

The Data Protection Act and you

By David Lewis

> "*Doublethink* means the power of holding two contradictory beliefs in one's mind simultaneously, and accepting both of them."
>
> *George Orwell, Nineteen Eighty-Four*
>
> "The member States of the Council of Europe [recognise] that it is necessary to reconcile the fundamental values of the respect for privacy and the free flow of information between peoples ..."
>
> *Preamble to the Council of Europe Convention for the protection of individuals with regard to automatic processing of personal data*

The Data Protection Act was passed by Parliament in the Orwellian year of 1984. The spirit behind it certainly seems to be libertarian and benevolent, and the Data Protection Registrar is far removed from Big Brother. But more later on the implications of the Act for civil liberties ...

The Data Protection Act has been phased in by stages: registration became compulsory in May 1986, and people's right to see data about themselves came into effect in November 1987.

The Act breaks new ground. Never before has the British computer-using community been subjected to such an elaborate legal code; and even having read this article you may still be wondering (like me) why it was necessary.

The process started with the Council of Europe Convention "for the protection of individuals with regard to automatic processing of personal data". The decision for the Government was whether or not to ratify the Convention. If we didn't ratify, the other European countries who had done might have made it difficult for their corporations to transfer personal data to firms here for commercial purposes. This would have damaged Britain's data processing industry, and that is why the Government felt obliged to introduce this Act.

"An Act to regulate the use of automatically processed information relating to individuals and the processing of services in respect of such information" is the Data Protection Act's 'long title'. The Act accomplishes these aims through a variety of measures, of which these are some:

- With some important exceptions, if you hold 'personal data' on a computer, you must register with the Data Protection Registrar. The register entry will govern the data you hold and the way you obtain, disclose and use it. (The Act quite correctly treats 'data' as a plural noun, incidentally, but that sounds pedantic to us.) All registration entries are open to public inspection for a small fee — both at the Registrar's office in Wilmslow, and (on microfiche) at major public libraries throughout the UK.
- You must comply with eight 'data protection principles'. If you don't, the Registrar can try to make you — either through 'de-registration' or by prosecution.
- You can be prosecuted — normally by the Data Protection Registrar — for a variety of new offences:
  - failing to register
  - obtaining, holding, or using personal data outside the scope of your registration
  - disclosing such data without authority outside the scope of the registration
  - transferring the data outside the UK beyond the scope of the registration
  - giving the Registrar false information
  - failing to comply with various Registrar's notices.
- The people on whom you keep data — the 'data subjects' — can insist on having a copy of it.
- A 'data subject' may be able to sue you for compensation if you lose, destroy or disclose the data, or if the data held is inaccurate.
- A court can order someone holding personal data on an individual to provide him or her with access to that data, and the court can also order that inaccurate data be rectified or erased.

It is fashionable to praise the Data Protection Act. After all, we have long been told by the great and the good that our civil liberties are about to be destroyed by the information revolution; it is natural that a piece of legislation which seems to protect those liberties should be welcomed.

I reject these modish assumptions. It seems to me an outrageous attack on civil liberties for a computer user — or anyone else for that matter — to be restricted in the keeping, use and disclosure of information, whether it happens to be stored in machine-readable form or on paper.

I do however favour a measure of subject access. I think, in other words, that individuals should have the right to see what data others are holding on them. And I also think they should have a limited right to amend alleged inaccuracies or to ensure that a corrective note is appended to the data.

But comprehensive registration along the lines of the Data Protection Act is not essential for these safeguards, and I don't think that the Council of Europe Convention really goes so far as to require this: there has perhaps been some excess of zeal on the part of the Parliamentary draftsmen in their endeavours to accord with the Convention.

And there is something about this Act that hints at a different kind of law, the one I call the Law of Unintended Consequences, which seems to operate whenever governments legislate in areas that are no concern of governments. In years to come it could lead to a certain diminution in those rights of free expression which we take for granted. And alongside that, I can foresee an imperceptible decline in the British and European data processing industries which we shall fully appreciate only when the Americans and the Japanese, unencumbered by such foolish laws, become the uncontested masters of the world of high technology.

Still, whatever my opinion (or yours) on the merits of the Act, we have to comply with it; and that is what this article is about.

Registering under the Act

You need to register if you 'hold personal data' or operate a 'computer bureau'.

As with nearly all Acts of Parliament, the statutory definitions are very important; and the way the Act defines 'data', 'personal data' and 'hold' is quite complicated. Further, there is a general exception relating to word processing plus a number of specific exemptions.

But you will need to review all your databases (interpreting that expression widely, so as to include for example data files used in conjunction with accounting software), spreadsheets, and any other files (other than ordinary text files used to produce documents) that contain or are likely in the future to contain information about living people.

In the box I suggest a simple procedure to help you decide whether, what and how to register. But if in doubt, do register. The cost is £40 for a three-year registration period, and that fee will cover *all* entries under your name (or your organisation's) — not merely the registration of a single use or purpose.

How do you register?

This article concentrates on computer users other than computer bureaux, and if you do run a computer bureau — that is to say, if you process information for other people or let them use your equipment — the registration process will be somewhat different from what is described here.

To register as a data user, you need the application form and explanatory notes in the 'Data Protection Registration Pack', available from Crown Post Offices or from the Registrar's office.

The application form is in two parts. Part A gives your name and address; Part B covers each purpose for which you intend to hold personal data. (There's a new simplified one-part form for small businesses, which you should ask for if appropriate.)

The forms give you multiple choices for describing such things as the purpose, the 'data subjects' on whom you're holding data, the data itself, and the likely or intended sources and disclosures. So completing the form shouldn't be too much of a problem.

It's a good idea to complete Part B widely enough to cover all likely possibilities, especially when indicating the categories of people to whom you might be disclosing the information. To enlarge on this a bit, the essence of the Data Protection Act (in somewhat oversimplified form) is that you're at liberty to do whatever you want with your data — provided that what you do is covered by the registration. So a little foresight and imagination may protect you from possible claims or prosecutions.

Having posted the application, you can proceed to hold, process and disclose personal data in accordance with it. But if the registration is refused, of course, you will have to cease holding the data; if registration is allowed subject to changes in your application, you'll have to alter the way you hold, process, use or disclose the data.

Rights of access

Anyone is entitled to be told whether you're holding personal data on them, and to be given a copy of the data.

So far I've assumed you're reading this article as an actual or prospective 'data user' — someone who holds and processes data, and may need to register. But in dealing with people's right to see information held about them on computer, it's easier for you to put yourself in the position of a 'data subject' who wishes to find out whether computerised information is held on him or her and to get at it. The data user is generally obliged to supply that information, which will normally be done by printing or perhaps by e-mailing it.

Should you register?

1. **LIST all the files you hold that might qualify**. Go through your disks and other media (including battery-backed RAM in portable computers and the like); and also access any remote computer which enables you to store files or which stores them automatically for you. As you do all this, you should note all files of the following types:

   - databases
   - spreadsheets
   - any other data files (which you can interpret widely, but you shouldn't include word-processed text files unless you're keeping them for some purpose other than simply for preparing text).

2. **DELETE from the list all files held only for one of the following applications**:

   - calculating pay or pensions (including deductions). But don't delete any payroll file which contains information over and above the strict needs of these calculations;
   - keeping accounts, or records of purchases or sales, either for business or for some other activity (so long as it's *your* business or activity)
   - keeping records of other transactions, for helping you to pay tax or to pay or collect a debt, or to make financial or management forecasts for your business or other activity.

3. **DELETE from the list any files held by an unincorporated members' club and relating only to the members of the club**.

4. **DELETE from the list any files whose contents and use you don't control in whole or in part**.

5. **MARK all files which you amend, add to, delete or rearrange on behalf of someone else**. These files should *not* be deleted from the list.

6. **MARK all files which are amended, added to, deleted or rearranged by other people on your equipment**. These files too should *not* be deleted from the list.

7. **DELETE from the list any files held by you as an individual** (and not as an organisation) which are concerned only with the management of your personal, family or household affairs, or held only for recreational purposes.

   Examples might be a database (in document form or otherwise) listing bulletin boards and their SYSOPS; a file maintained for insurance purposes that records the names of the family members owning each item in your home; or a database of your books or records, listing the authors, composers or artists (including of course persons still living).

   On the whole, you're unlikely to have to register these — unless you keep them for some non-personal, non-family, non-household, or non-recreational purpose.

8. **DELETE from the list any files held only for the purpose of distributing (or recording the distribution of) articles or information**, and consisting only of names, addresses or other particulars necessary for making the distribution.

9. **DELETE from the list all permanently read-only files** — to be more precise, those in which the names or other identifiers of living individuals can't be amended, added to, deleted or re-arranged — or haven't been and aren't intended to be so altered — either by you or by anyone else on your behalf.

10. **DELETE from the list all files which contain no reference to living individuals** (whether by name or in some other way enabling you to identify them).

11. **DELETE from the list any files which merely indicate your intentions towards the individuals concerned** and don't actually contain any information or opinions about any of them.

---

12. **IF THERE ARE ANY MARKED FILES LEFT IN YOUR LIST** you will need to register as a **computer bureau**.

13. **IF THERE ARE ANY UNMARKED FILES LEFT IN YOUR LIST** you should register as a **data user**. And you may well need to complete a separate application for each of those files (unless two or more cover the same purposes).

If you were seeking this kind of information yourself, you should first inspect the register (at major public libraries, where microfiche copies are kept) to see whether the person or firm concerned is likely to be holding personal data about you. The Act doesn't say in as many words that you must refer to a register entry when seeking access, but I think you would run into difficulties if you didn't and if the data user had registered under more than one purpose.

You might then write to the data user along these lines:

> ' I refer to your register entry in the Register of Data Users and Computer Bureaux under registration number *nnnnnnnn*, and in particular to Purpose *nn* of that entry.
>
> ' I hereby request you, under section 21 of the Data Protection Act 1984 (a) to inform me whether the data held by you include personal data of which I am the data subject; and (b) to supply me with a copy of the information constituting any such personal data held by you.
>
> ' I enclose the statutory maximum fee of £10. Please refund any balance if the fee which you require for the supply of this information is less than £10.'

You must make a separate written request for each of the relevant purposes specified in the data user's registration and likely to relate to the information you're seeking.

Note also that the data user can ask you for proof of identity and for enough further information to locate the information you're seeking.

The data user must comply with your request within 40 days of receiving the request or (if later) the proof of identity and further information mentioned.

The Data Protection Principles

The Act lays down eight 'data protection principles', and the Registrar won't allow your application if it looks as though you're proposing to breach any of them. Further, if you do breach them after registration and the Registrar finds out about it, he can take legal steps to make you comply with them (including the threat to de-register you).

The principles can be analysed into those which are based on the registration and those which aren't. The registration-based principles require the data user ...

- to hold personal data only for a purpose which is registered
- not to use or disclose the personal data except in line with the registration
- to ensure that the personal data itself is 'adequate, relevant and not excessive' in relation to those registered purposes
- to keep the data no longer than is necessary for those purposes.

The other principles aren't registration-based:

- Personal data must be obtained and processed 'fairly and lawfully'. This means you mustn't mislead the people from whom you get the data as to the intended purpose (though you needn't necessarily state the use or disclosure if the personal data is for historical, statistical or research uses).
- Personal data must be held only for lawful purposes. It must be accurate and, where necessary, kept up to date.
- A further principle entitles individuals to be told by a data user whether the data user holds any personal data about them, and to be given access to such data. (This is virtually duplicated by the subject access provisions I've already covered.)
- The eighth and last principle requires 'appropriate security measures' to be taken against unauthorised access to, or alteration, disclosure or destruction of, personal data, and against its accidental loss or destruction. **'Disclosing' data includes disclosing information extracted from the data, and this means computer printout.**

The need to avoid the unauthorised or accidental destruction of personal data is interesting, as this seems to be a matter affecting the interests of the data user rather than the data subject. But there it is; and to comply with it you will have to keep back-ups of your disks — and you may well need to take further measures, such as storing them away from the originals.

Your security measures need to be serious and comprehensive. For example, you should not overlook the confidentiality of the monitor. Many people who enter my office wander behind my desk while I'm busy reading whatever they've put in front of me, and like flies to a jamjar their eyes are drawn to whatever happens to be on the screen of my terminal. I try to stop this practice (to which I confess to be no less prone), and I've also become deft at quickly applying the keystrokes which blank the screen.

Not only this, but so far as all personal data is concerned you must make a habit of locking away your disks and computer printout, and locking your office when absent from it — especially when the computer is on and the disks or printout are not shut away.

Other useful security measures include write-protecting sensitive files, and/or 'hiding' them so

that they won't appear in a directory listing. (To do this, you can use a public domain utility like SWEEP, ALTER or CHMOD; or the Norton Utilities' FA.EXE). A further protection is encryption, perhaps using the public domain ARC.COM. *(All these are available at very modest cost from our* DEALS FOR READERS *mail-order service; drop us a line for details, or call 01-833 3501 — Ed.)*

Prosecution

Under the Data Protection Act, you can be prosecuted (normally by the Registrar) for a number of offences. They include:

- 'knowingly or recklessly' obtaining, holding, using, disclosing or expatriating personal data without or inconsistently with the registration
- providing false or misleading information to the Registrar.

The penalty on conviction is an unlimited fine. Though unlikely to be applied rigorously in the case of a well-intentioned person or company, this fearsome sanction ought to encourage compliance with the law; I would strongly urge you to make a good start by forthwith registering any holding of personal data that might fall within the Act.

As for civil remedies, the data user can also be sued for damage and accompanying distress that is suffered because personal data is inaccurate, lost, or destroyed or disclosed without authority.

So don't be a law-breaker. The Data Protection Registrar has limited resources, and has no easy way of knowing that you haven't registered when you should unless someone tells him. But someone may well tell him — perhaps someone on that list of names and addresses you use for your mailshots, or someone who knows you're a business computer user and would like to make trouble for you. Take action now!

The Act for comms users

A typical comms user subscribes to some public services — Prestel and MicroLink or Telecom Gold, say — and may log on to various bulletin boards. The chances are that at some time such a user will be 'holding' personal data within the meaning of the Act — not just on your own personal computer, but actually on the remote machines used by the system providers.

This is clear from the part of the definition which says that you 'hold' data if (among other things) you control its contents and use it either alone or — and this is the point — jointly or in common with others.

This would cover the contents of your various mailboxes, which you can delete at will on most systems. The same would apply to messages uploaded to bulletin boards, since you can usually delete them too.

Are you then liable to register the purpose for which you use your particular e-mail or BBS facility?

The safe answer is *yes*. That's partly for various reasons involving legal technicalities, but also because the Data Protection Registrar certainly contemplates the registration of electronic mail subscribers.

Suppose then that I decide to register on the basis of my correspondence with the Editor about this article. How do I complete the application form? I'll give some brief details about this, which you may like to keep in case they help you with your own application.

First, I have to pick a purpose, starting with the Data Protection Registrar's 70 standard purposes. Unfortunately, none of them quite fits this correspondence, so I shall have to describe the purpose in my own words — such as 'writing magazine articles'.

Then I have to describe the data subjects — with a list of 40 types to choose from, and a further choice of 'current', 'past' and/or 'potential' against each of them! As your Editor and I are the only individuals about whom information is likely to be held (our mailbox numbers, if nothing more), I shall tick the boxes for 'S003 Employees of other organisations — current', 'S004 Recipients, customers or clients for goods or services (direct or indirect) — current', 'S012 Business or other contacts — current', and 'S024 Correspondents and enquirers — current and potential'. That should cover the Editor, and anyone else who may write to me about this article. (Comments and queries are welcome, incidentally; contact me c/o The Editor, or via Telecom Gold mailbox 72:MAG20385.)

Now I must indicate what kind of data I shall be holding. Again, there are lots of boxes to choose from. The only information I expect to be holding on the Editor is his mailbox number; but to be on the safe side I shall tick against 'Personal identifiers C001', 'Identifiers issued by public bodies C003', 'Personal details C011', 'Business activities of the data subject C101', and 'Agreements, contracts C102'.

The next problem is 'sources and disclosure'. I'll mark 'The Data Subjects themselves D101 — Source and Disclosure', and maybe a few others such as 'Employees, agents D105 — Source and Disclosure' — if in doubt, tick it.

This doesn't bring us to the end of the form, but I think it's as far as I need go in this article. You probably get the idea: don't tick any box that couldn't be relevant, but try to tick all the boxes that could. Don't ponder and agonise if you're not sure whether to tick a particular box: if in doubt, just tick it, and you won't go far wrong.

Examples

It may also help if I go on to show you one or two more concrete examples of typical (or at least possible) comms usage, and see whether you would need to register in each case.

Case 1 is where you have used your word processor to compose a message for a BBS. The message is about some interesting novels you want to recommend to other readers of the bulletin board. You log on and upload the message. (Let's assume it stays in the host computer for a few months, and that the original word-processed text file isn't deleted immediately. So you're 'holding' data for some time at both ends of the phone line.)

Now the first point — which is easily overlooked — is that your list of novels no doubt refers to the names of the authors, quite a few of whom may be still living. This is 'personal data' all right. But you only have to register if you're 'holding' it; so *are* you holding it?

Essentially you're 'holding' data if you have 'processed' it or if you intend to. 'Processing' too is elaborately defined, and the chances are that you *did* process the data when you composed the document using your word processor ... though the definition does say that for processing to take place it must be done 'by reference to the data subject'. So that's the next question: when you 'processed' (amended, augmented, deleted, rearranged) your text, you did so 'by reference to' Kingsley Amis and your other favourite living authors?

The Data Protection Registrar says you did indeed 'process' your text by reference to Kingsley Amis etc if you did so because it related to those writers. On the whole, it seems unlikely that your list of works was centered around the authors themselves, in the sense of the Registrar's interpretation. But you decided to do a search and replace operation in the course of creating the document, substituting (say) 'Amis, Kingsley' for 'Amis, K.', then you would have been 'processing' personal data; and you would therefore be 'holding' it.

Fortunately there are two exemptions which probably ensure that you don't need to register for this purpose alone. The first says you aren't 'processing' data if you amend, augment, delete or rearrange it solely for the purpose of preparing the text of documents. And the second exemption is that you don't have to register, or afford subject access facilities, if your personal data is concerned only with your personal, family or household affairs, or is held for recreational purposes.

Case 2 sees you downloading a directory of users subscribing to an e-mail or bulletin board system. You intend to print it, but also to keep it on disk in order to search for names and other text.

Apart from any minor editing that might be involved, there would be no 'processing' in the Data Protection Act's sense. It's the *intention* that counts — you don't *intend* to add, delete, alter or re-order any of the entries; and read-only operations don't constitute processing. So it wouldn't be necessary to register, even if you intended to use the directory for business purposes.

In Case 3 you're creating a file of names and telephone numbers for use with the automatic dialling facility of your communications software. Those names include living people, both friends and business contacts. Some of the data, then, constitutes 'personal data'. You certainly intend to amend and add names to the file, so you're 'processing' the data. Further, you're doing so by reference to the data subjects themselves, since the telephone numbers are attached to the names rather than *vice versa*. You are therefore clearly 'holding' the data.

Do any of the relevant exemptions apply? The file isn't concerned only with the management of your personal, family or household affairs, or held only for recreational purposes, because the data includes business contacts. It may be held only for the purpose of 'distributions' or recording distributions: but it would be wise not to rely overmuch on that exemption, which is a bit unclear.

So you may need to register this purpose — for instance as 'P004 Marketing and Selling (including Direct Marketing to Individuals', or 'P007 Management of Agents and Intermediaries', coupled with 'P012 Ancillary and Support Functions'; or perhaps describing a special purpose in your own words.

Last words

For any computer user the best practice is to register, to include as many purposes as possible in the application (it won't cost you any more than the basic £40 registration fee), and generally to complete the forms on the principle of covering all the possibilities you can think of. ◆

Now read on

The Office of the Data Protection Registrar publishes a lot of helpful (and free) material. Ask for a Data Protection registration pack and for the eight *Guideline* booklets.

These you can get by telephoning the Registrar's Enquiry Service — **0625 535777** — or via electronic mail (Telecom Gold 74:CKK057). You can also write to the Office of the Data Protection Registrar at Springfield House, Water Lane, Wilmslow, Cheshire SK9 5AX.

The Registration Pack can also be obtained from main Post Offices, and the Registrar has been taking stands at many computer shows with all the bumf available there as well.

A Turbo Pascal Primer

Part Four: Data types

By Mark Whitehorn

For programmers reared on Basic, one of Turbo Pascal's more annoying features is the use of *data types*. Instead of being allowed to stuff any sort of data into a variable, you are required to tell Turbo what variable names you'll be using and what kind of data is going to be crammed into them. If you are subsequently foolish enough to put any other form of data in there, the wretched program is liable to crash.

Why does Pascal require you to declare the variables at the start of the program?

Niklaus Wirth (who developed the original Pascal) had problems with his students writing messy, unreadable code. By forcing them to declare the variables they would be using at the start of the program, he felt that their programs would be more readable.

He also found that students often misspelt the name of a variable in the body of the program. In languages such as Basic this is disastrous — the program continues to run using a *new* variable which has a value of zero. This type of bug can be particularly difficult to find. Turbo Pascal will pick up all your misspellings, because it thinks they are undeclared variables — which in fact they are. So you never have to chase this type of bug. I regard this as one of the language's great advantages, primarily because my spolling is so poor.

But why does Pascal have different data types? Why doesn't the language just allow you to declare the variable names at the start of the program, without bothering with lots of data types? The answer is twofold.

- Restricting the variables to particular data types saves on memory. For example, holding a variable of type Char takes one byte of memory; a variable of type Integer takes two bytes.
- The compiler can compile code faster, and the resulting program will execute faster, if the data types are properly declared.

However it is worth bearing in mind that as memory becomes cheaper and micros become faster, these points becomes less important. And clearly neither of these considerations is important if the number of variables is small; but using an array of 3,000 integers rather than 3,000 reals makes a significant saving in microcomputer terms.

Variable conversion

While we're on the subject of data types, let's look at variable conversion. There are occasions when it is imperative to transfer information from one data type to another.

Conversion from Integers to Reals is perfectly straightforward. Suppose you have a number in an Integer variable called *IntegerNum* and you wish to transfer it to a Real variable called *RealNum*. This statement will be perfectly acceptable to the Turbo compiler:

```
RealNum := IntegerNum
```

On the other hand, the converse form ...

```
IntegerNum := RealNum
```

... will crash the program, even if the number in *IntegerNum* happens to be a whole number.

The transfer can be done, if you use the function *Round* or *Trunc* to remove the numbers after the decimal point.

Round rounds a number to the nearest whole number.

Trunc chops off anything after the decimal point.

Turbo Pascal provides two further type conversion functions:

Chr converts a number (which must be of type Integer, not Real) into a character whose ASCII code is the same as the number.

Ord converts a character (type Char) into a number (Real or Integer) which is the ASCII code of that character.

The use of these four functions can be demonstrated with the example program following:

- This program takes in a character, converts it into the ASCII code of the character and stores that number in variable of type Real. This number is then written to the screen
- 0.23134 is then added to the number to ensure that it genuinely is a real number (and not an integer in heavy disguise)
- this real is converted into an integer using Trunc (Round works just as well here). The number is again written to the screen
- finally this real is converted into a character again and written to the screen

- the program will repeat until a letter is pressed whose ASCII code is 113. (Guess which character has this code.

```
Var
Ch,
NewChar     : Char;
RealNum     : Real;
IntegerNum  : Integer;

Begin;
ClrScr;
Repeat
 GotoXY(1,5);
 Write('Please enter a character. ');
 Read(Kbd,Ch);
 Writeln(Ch);
 RealNum := Ord(Ch);
 Writeln('The ASCII code of this number is ...',
 RealNum:5:0);
 RealNum := RealNum + 0.23134;
 IntegerNum := Trunc(RealNum);
 NewChar := Chr(IntegerNum);
 Writeln('Here it is as a character again ...',
 NewChar);
Until IntegerNum = 113;
End.

{P.S. Always use Read(Kbd, ...) rather than
Readln(Kbd, ...)}
```

I/O error-checking

Users of our programs often make mistakes entering data into the programs which we write. A common one is to attempt to enter the wrong type of data into a variable. You have asked for a numeric input, reading into a real; and the user inputs his name. Result — the program crashes. Similarly the program in the last episode, which read from a file, produced an error if it couldn't find the correct filename on disk.

Rather than have the program crash about the user's ears, it would be better if your program detected the problem and produced a helpful message to guide the user on the correct course of action. Happily, this is possible.

Turbo Pascal allows you to tell the compiler not to perform its own error-checking and instead leave it up to the program and you. That's done using a 'compiler directive', a direct instruction to the compiler. In this case we want to tell it not to perform its own I/O check, and the directive to use is {$I-} . (In a Turbo Pascal program, almost anything inside curly brackets is regarded as a programmer's comment. The only exceptions are the compiler directives, which have a dollar sign immediately after the opening bracket.)

Now checking for I/O errors is your responsibility; but Turbo Pascal will still give you some help. After every I/O operation you can have a look at the built-in integer function *IOResult*. This is normally set to 0; but after an I/O operation which was in error, it has a value of 1 or greater. So to produce error-free input, try something like this:

```
Var
Age : Integer;
Begin;
{$I-}    {Turns off the error-checking}
Repeat
 Write('Please enter your age. ');
 Readln(Age);
Until IOResult = 0;  {$I+}
End.
```

That repeatedly asks for the user's age until he or she enters a number. This sort of thing works very well; and it can be tidied up a little with some of Turbo's built-in procedures:

```
ClrScr;
Repeat
 GotoXY(5,5);
 ClrEol;
 Write('Please enter your age. ');
 Readln(Age);
Until IOResult = 0;
```

Rather cleverly, simply calling *IOResult* causes it to be reset to 0 after it has returned a result. This ensures that the next time an I/O is attempted, *IOResult* will return a 0 unless another error has occurred.

This section of program is fine as far as it goes. But it doesn't tell the user what *kind* of input fault has occurred. Well, in fact the value of *IOResult* can indicates the type of error which has occurred; and we can use this to tell the user what went wrong.

This is well documented in the *Turbo Tutor* on pages 228 to 230. With Borland's kind permission I reproduce here a slightly adapted version of the I/O checking procedures from the *Tutor*, incorporated into a (basically useless) program which demonstrates how the procedures are used.

I've included an inordinate amount of documentation within the program, but it seems more sensible to put it there rather than in the body of the article. (I mention this only because a friend came along while I was writing the program and poured scorn on it. Nuts to him, better too much than too little.) In addition I have made a conscious effort to structure the program properly, as an example of the principles discussed in Part Two. Again, this is a little unnecessary in a program of this size: but in larger programs, layout of this type is invaluable. ◆

```
{Note: This program demonstrates how to write I/O routines which don't crash
the program on error. The program looks for a file to open. It will repeatedly
ask for a valid file specification until it gets one. So it is a good idea to have
ready on the disk a little file (perhaps PENGUIN.TXT from last episode). You can
then try invalid filespecs to see what happens on error, and then enter
PENGUIN.TXT to allow the program to continue.}

Var
 Age                    : Real;
 FileSpecification      : String[30];
 NameUsedInsideProgram  : Text;
 IOErr                  :Boolean;
Type
 Prompt       =  string[80];
{-------------------------------------------------------------------------}
{                          LEVEL THREE PROCEDURE                          }
{-------------------------------------------------------------------------}
Procedure Error(Msg:Prompt);
{A bit of Whitehorn documentation here. A string (which contains the
error message) is passed to this procedure from IOCheck. When the string arrives
it is transferred to the local variable Msg. Thus the line
        Write (^G,Msg,'  Press any key to continue');
makes a noise (the ^G), prints the error message, and then asks for a key
to be pressed. The read(Kbd,Ch) was covered in the previous installment,
and just waits for a key to be pressed.}

Var
 Ch : Char;
Begin;
 GotoXY(1,20); ClrEol;
 Write (^G,Msg,'  Press any key to continue');
 Read(Kbd,Ch)
End; {of proc Error}

{-------------------------------------------------------------------------}
{                          LEVEL TWO PROCEDURE                            }
{-------------------------------------------------------------------------}
Procedure IOCheck;
{Me again, with a little more documentation. This procedure is called after
each I/O operation that is attempted with the compiler set to $I-. The
procedure first sets a local variable called IOCode to the value in
IOResult. This is necessary since a single call to IOResult resets IOResult
to 0, so we have to capture the number therein first time around. With the
number from IOResult captured in IOCode, we come to the line:
        IOErr :=( IOCode <> 0 );
This means "Set the boolean variable to TRUE if IOCode is not 0" — in other
words, "set IOErr to TRUE if an error has occurred on the last I/O
operation". The line
        if IOErr then begin
ensures that the Case...of only operates if there has been an error. The
Case...of ensures that the correct error message is written out by the
procedure called Error. Note that the procedure called Error is passed a long
string which is the error message.}

Var
 IOCode :integer;
Begin
 IOCode :=IOResult;
 IOErr :=( IOCode <> 0 );
 if IOErr then begin
   case IOCode of
    $01 : Error('File does not exist');
    $02 : Error('File not open for input');
    $03 : Error('File not open for output');
    $04 : Error('File not open');
    $10 : Error('Error in numeric format');
    $20 : Error('Operation not allowed on logical device');
    $21 : Error('Not allowed in direct mode');
    $22 : Error('Assign to standard files not allowed');
    $90 : Error('Record length mismatch');
    $91 : Error('Seek beyond end of file');
```

```
    $99 : Error('Unexpected end of file');
    $F0 : Error('Disk write error');
    $F1 : Error('Directory is full');
    $F2 : Error('File size overflow');
    $F3 : Error('Too many open files');
    $FF : Error('File disappeared');
   else
    Error('Unknown I/O error:  ');
    Write(IOCode:3)
  End {case}
 End;
End;{of Proc IOCheck}

{-------------------------------------------------------------------------}
{                          LEVEL ONE PROCEDURES                           }
{-------------------------------------------------------------------------}
Procedure GetTheUsersAge;
Begin;
 {$I-}
 Repeat
   IOErr := False;
   ClrScr;
   GotoXY(5,5); ClrEol;
   Write('Hi. Please enter your age.   '); Read(Age);
   IOCheck;
 Until IOErr = False;
 {$I+}
End;   {Of Procedure GetTheUsersAge}

{-------------------------------------------------------------------------}
Procedure GetTheFileToOpen;
Begin;
 {$I-}
 Repeat
   IOErr := False;
   ClrScr;
   GotoXY(5,5);
   Write('Please enter the name of a file to open.   ');
   Read(FileSpecification);
   Assign(NameUsedInsideProgram,FileSpecification);
   Reset(NameUsedInsideProgram);
   IOCheck;
 Until IOErr = False;
 {$I+}
 ClrScr;
 GotoXY(5,5);
 Writeln('File open.');
 GotoXY(5,15);
 Write ('Press any key to continue');
 Repeat until KeyPressed;
End;  {Of Procedure GetTheFileToOpen}

{-------------------------------------------------------------------------}
Procedure CloseTheFile;
Begin;
 {$I-}
 Repeat
   IOErr := False;
   ClrScr;
   GotoXY(5,5);
   Write('Closing file.');
   Close(NameUsedInsideProgram);
   IOCheck;
 Until IOErr = False;
 {$I+}
End;   {Of Procedure CloseTheFile}

{-------------------------------------------------------------------------}
Begin {Main Program}
  GetTheUsersAge;
  GetTheFileToOpen;
  CloseTheFile
End.
```

Free Small Ads

Apricot Sales

■■■ FREE: Symphony, Lotus 1-2-3, WordPerfect, Open Access, Smart, SuperCalc 3, Logistix, Psion Xchange, Word, Cardbox Plus, DBman, ScannerS, and thousands of other programs, utilities etc. Not to mention every Doodles ever written (in the original unedited version) ... provided you fork out £1,299.95 for the author's own Xi10. Actually this is now an Xi20 with 960K of RAM. Call David St.John-Wallis on 0428 712462.

■■■ APRICOT Xi 10MB hard disk, 720KB floppy, 512KB memory. Little used. Best offers around £650 plus VAT. 01-622-5887.

Apricot Add-on Sales

■■■ ALCOR MULTI-BASIC COMPILER for MS-Basic. Allows long lines, labels, etc. Two s/s disks, 200-page manual. £40. David Ryder, 0204 43910.

Other Sales

■■■ CANON PW-1156A printer needs good home. Used by one-man-band chartered accountants. £150 plus VAT. Phone Sevenoaks 62396 9am-8pm.

Help

■■■ F SERIES user wants to get in touch with others in the States. Contact Bob Mandel, Lewis & Clark College, Box 96, Portland, OR 97219, USA.

■■■ PORTABLE user would like to contact another in South London to exchange information. Contact Geoff Reade, 9 Taymount Grange, London SE23 3UH.

Services

■■■ DISK RECOVERY: problems with deleted files, scrambled hard disks, and the like? We can (probably) help — no fix, no fee! Call Roger Gann at TP Group on 01-833 3501.

■■■ FILE TRANSFER: we can convert files between Apricot single- and double-sided, IBM PC format, PC/AT format, and Sirius — cheap! Call Roger Gann at TP Group on 01-833 3501.

■■■ HELP for PC/Xi/Xen users — setting up systems, MS-DOS, directory organisation, comms, modems. ALSO utility and business software written (in C, assembler or Basic). Call Tony Bridges, 01-948 1603 any time.

■■■ REPAIRS guaranteed. Martin Filmer Computing Services. 01-761 0425.

■■■ A0 PLOTTING — time on a Calcomp A0 plotter available for rent: Sirius or Apricot disks plotted on your paper or ours. Terms negotiable, but cheap. Phone James McEvoy on 01-859 4014 or 01-852 7981; or write to 16 Dunvegan Road, London SE19 1SA.

■■■ DESKTOP PUBLISHING: we can produce your newsletters, price lists, press releases etc in near-typeset-quality on our laser printer using our DTP software NEWSWRITER from your disks in virtually any format produced by word processor and/or computer. Fast turnaround, guaranteed quality, low prices. You're reading an example of our work right now ... Call Roger Gann at TP Group on 01-833 3501.

Wants

■■■ URHENTLY REQUIRED — F10 to join two others. Good price paid for a unit in good condition. Phone David Sparks on 0734 62787 or Telecom Gold 81:TWH167.

■■■ APRICOT PC with monitor and twin 720K drives wanted — would consider an Xi. Call James Gray, 01-839 6651 (day).

■■■ APRICOT PC TECHNICAL REFERENCE MANUAL wanted. Phone Simon Protheroe, 01-836 5454 ext 2161 (office hours) or 01-263 4489 (after 6pm).

■■■ APRICOT PORTABLE wanted, 512KB. Also SOFTWARE for FP and PC. Phone 0423 521439 or write E Holgate, 18 Montpellier Parade, Harrogate, HG1 2TG.

■■■ WANTED — Keypad for F2 (even a dead one will do). Robert Clews, Flat 6 Lindsworth School, Monyhull Hall Road, Kings Norton, Birmingham B30 3QA.

■■■ WANTED — dead/damaged/cheap Xis or PCs. Phone Derek on 01-650 4900.

■■■ APRICOT XEN (non-IBM-compatible) hard disk system. Also wanted — a good C INTERPRETER (to avoid having to link after every re-edit) or Apricot CODEVIEW debugger; and a TELETEXT TUNER/READER that can be connected to Apricots (is there such a thing?). Call Tony Bridges, 01-948 1603 any time.

■■■ F10 10MB hard disk system (with manuals if possible). Contact R Noakes, 051-336 6898.

■■■ IBM-COMPATIBILITY FOR F SERIES — either a fully-effective IBM emulator or a patch permitting the IBM version of Microsoft Word 3.0 to run. Also wanted — a WORLD MAP or GLOBAL MAPPING PROGRAM. I have scores of home-grown GW-BASIC programs to sell, exchange or share — including full documentation and sample programs for the infra-red Mouse. Contact Bob Mandel, Lewis & Clark College, Box 96, Portland, OR 97219, USA.

■■■ SECONDHAND OR EX-DEMO HARD DISK — internal or external, 10 or 20MB, for PC. Cannot afford £800+ for a new one. Write to T Orekoya, FAO (FIRI), Via della Terme di Caracalla, 00100 Rome, Italy.

■■■ MMS INSTRUCTION MANUALS AND INTERFACE BOARD required for MMS hard disk unit and tape streamer. Willing to pay for help (even telephone assistance) in setting this up. Call Roy Roberts, 0923 775381 (office) or 0494 775088 (evenings).

■■■ GOOD DEBUGGER wanted for PC/Xi. I hate using Apricot's; can anyone help before I'm forced to write my own? (Preferably with symbols and/or C source display). Phone 01-948 1603 any time.

■■■ 10/20MB EXTERNAL HARD DISK required for Apricot PC with BIOS 2.4. G S Jayaram, PO Box 2642, Doha, Qatar (phone 417282 6-9pm UK time).

■■■ APRICOT PC or 10MB XI wanted; also PRINTER if available. Fair price paid, immediate cash. Phone Mrs Morrison, 0636 704208.

■■■ WORD PROCESSOR wanted for Apricot PC with single-sided disks — MS-WORD 3.0 preferred. Also ANY OTHER SOFTWARE. Phone 01-908 4631

Use the service!

The Classified Ads service is free to all readers of APRICOT FILE. Just let us have a sheet of paper with the words (not too many); your ad will appear at least once in the newsletter and for around three months in the DEALS FOR READERS broadsheet.

Send your ad to us via electronic mail (Telecom Gold 81:JET767) or by land mail to:

APRICOT FILE
TP GROUP
PO Box 509
London N1 1YL

Doodles by David

Tips and Tricks

The occasionally essential column

By David St.John Wallis

A perfect demonstration of cause and effect has been noted by statisticians from Kings Cross to the outer moons of Alpha Centauri in their analyses of cancelled APRICOT FILE subscriptions and the absence of the *Doodles* column. Sorry folks. I was away for a long time, changed house and spouse, bought a clone etc. Now you can get your poison pens out. A Pretend Author gets to work again.

Remorse

Certain fellows receive hospitality from certain cricketers in far flung Singapore, 32°C and getting wetter. And how do they repay this, I hear you howl. Well, they go and erase their vastly complicated AUTOEXEC.BATs — that's what they do. Is it any compensation that said punter hands over real American utilities like TORNADO NOTES and MACE and PCTOOLS and SMARTNOTES which might work? No, it isn't.

Remorse, thy name is Deviate. As I just may have remarked before in these pages, these things are sent to try us. Or in this case, they are sent to try the patience of gentle Botham-lookalike Ian Craven who runs the Far East with a rod of iron. Ian, however, was God-like in his patience. He forgot (or just decided not) to send round the Sumo wrestlers to the Pan Pacific hotel to rearrange this columnist's equipment. What a nice man. Need to wreck a computer without complaint? Call Ian in Singapore. He knows how to forgive a chap. Even a chap who drank all his vodka.

The moral in this trivia? COPY AUTOEXEC.BAT DREADFUL.BAT.

Faster and faster

Spreadsheet buffs who noted the item in APRICOT FILE Release 2.4 about spreadsheet speed might like to know how Lotus 2.1 performed on an Apricot 80286 machine. Using exactly the same test, and without a numeric co-processor, this spreadsheet took slightly less than 4.9 seconds to complete all 1,524 calculations. The end result was -1.17935, or -1.179356 with extra column width. And I don't know why Lotus didn't round up either. In any case, the result was as good as makes no difference.

The relevance of all? Well, an Xi20 using SuperCalc 3 did the same test in 183.2 seconds. So this combination of hard and software was *37 times* quicker.

In the fullness of time I hope to run the test on an 80386 with a co-processor. But for now, this is fast enough. (For afiçionados of this kind of thing, the derived file size was 47,736 bytes normally; and 13,627 — less than a third smaller — after auto-squeezing.)

What's all this to me, I hear you snivel. Well, a certain Rxgxr Gxnn has obtained from the back of my Reliant Robin a Lotus 1-2-3 V2.1 which works on Apricots. Wow!

Two tips!

My accountancy friend who wields a mean quill from Rishangles is none too delighted with the verbose Glen Siegel messages in our KOPY utility. But try this:

```
KOPY A:*.* B:/P/L
```

This is a bit dangerous, though — the 'L' suppresses the error messages too. Helpful things like "Error code 6, invalid handle" and "Goodness, the teapot is leaking again".

Another tip, this one using the Norton SA screen attributes command (I thought of this all by myself). Call it BLANK.BAT and try it.

```
SA BLACK ON BLACK
PROMPT $h
ECHO OFF
CLS
PAUSE
SA BRIGHT GREEN ON BLACK
CLS
PROMPTD
```

And PROMPTD goes ...

```
PROMPT $d$h$h$h$h$h $t$h$h$h$h$h$h $p $e
```

The Clone File

If you like this next bit, and Dennis likes it too, you'll get more; impressions of working things on clones which until now you worked on Apricots.

I thought I might start with The Big Switch. I've got to admit it — I finally gave up on the Model T (see the Small Ads page). And all my friend had a clone, and I had to do some clone work, so a clone is what I had to get.

Should *you* switch? The balanced view would be 'no', if you're happy with what you've got. But if you're unbalanced, and actually like computers, then going clone means you can run practically anything. And lots of the software is either free or (thanks to A.Sugar and wily Orientals) cheaper than Apricot versions. And wonderfuller too.

Most clones are faster than most wrinkly Apricots, too, and it's not true that processor speed

makes no difference in word-processing. It is nice to be able to replace every occurrence of marginal re-investment potential with thundering great losses in a second or two in an 800-page report without taking a holiday in Reading while your original pre-Cambrian cpu is chuntering on with the job. If you're the kind of chap who still goes to work on a bike in mid-winter, this argument won't cut much ice. But you've probably got a nice comfy car, haven't you? Which is quick? And warm? Quick, warm and comfy is how you get to feel about your clone after you've learned its little ways.

And there are lots of little ways to learn. Like a rotten type-ahead buffer. Messages which say things like 'fatal error on drive C' and 'you must be an incredible chump to do something like that'.

This article is addressed mainly at those fellows who are thinking the way I did. My clone is an Apricot 80286 running at 10mHz; my friend got an Amstrad and it doesn't run lots of things. So I'm not going to talk about clone-compatibility with IBM, rather more about the *Apricot* Xi and *Apricot's* IBM clone.

The first thing, of course, is what's agonna run. Well, BRAINSTORM did and lots of utilities and one or two other important things like most of ScannerS (shortly to be re-issued as SpannerS for clones. Puff). But bigger programs like Lotus and CardBox and PARADOX and SuperWriter need to be in clone format. (Curiously I received a version of SuperWriter which works well. Also there's a SuperWriter 2 out which looks totally different). Incidentally, this article is being written on Lotus Manuscript — which is a whole lot better than WordPerfect, because you don't have to remember that every function key on a clone has no less than *four* meanings, and it doesn't keep bothering you with maddening hyphenation messages following file imports or text retrieves or whatever crazy jargon they dreamed up in 4.2 which hopefully they'll sort out in WordPerfect 5 (coming Real Soon Now).

This whole thing is actually a big subject and perhaps I'll deal with it more later. And properly. (Fat chance, Ed). The main thing you'll want to know, won't you, is will you be able to read your wrinkly data. Yes you will. Provided that: for word processors you either get a clone version of the Apricot one, or you save all the wrinkly material in ASCII, non-document, non-WP, format.

You will also need a facility for translating things from little disks to bigger ones. Chaps in the know use TRANSEND, which does the business a treat. Insert an appropriate piece of cable into both computers (get advice on this bit — it's sexy). Shove a TRANSEND disk into both machines and load the TRANSEND program at both ends by typing "TRANSEND".

At one of the computers you type a command like

```
D C:\DATA [Return]
```

... and then ...

```
E A:\DATA [Return]
```

This tells the computer you're typing at to use directory \DATA on the C: drive and that it's going to be looking at the \DATA subdirectory on the A: drive of the other one. Then then you go something like G *.*, and everything from A:\DATA is fed across the wire to C:\DATA.

You need to be a trifle careful about porting across things like COMMAND.COM into root directories. Well, you knew that didn't you?

(Actually, there are other ways — use a disk conversion service like ours, with prices from £10 for the first disk and less thereafter. Or buy a clone with 3½in disks, and use MS-DOS 3 on your Apricot to save files in IBM *format. Or use Telecom Gold as a file transfer medium for* ASCII *files — Ed.)*

The tough bit about all this is getting the program and the piece of cable. Oh no it isn't, quoth Ed through his beard. Look in the DEALS FOR READERS list. (Do you know I don't get paid. Other writers do. Not me. I do it for love or ego-massage or something. And contrary to contrary views, I hardly ever sell anything in consequence of these *Doodles*. More likely that I wind up giving things away. So let there be no more snide comments about chaps trying to sell their stuff sotto voce and subliminally through these columns.)

Sorry about that. Back to the point. Having got the programs and data into the clone, you should be able to load ...

- word processing data in ASCII into any word processor
- 1-2-3 files into 1-2-3 (or via a translate utility into another spreadsheet)
- database material into 1-2-3, or another database which has an import capability

... and so on. Then you just have to get to grips with the keyboard, where many of the keys seem to be in the wrong place; and a new version of the operating system, probably (though usually they get better as they get into higher numbers — I'm using 3.2, which has clever SUBST and XCOPY and REPLACE utilities).

There'll be more of these Apricot-to-clone rantings if you want them. Send money or other indications.

Yankee Doodles

- 80386 'power' computer reviews are all the rage in the States right now. But none of them include the Apricot. Woe!
- In their various categories here, the top programs are WordPerfect 4.2, dBASE 3+, 1-2-3 and Harvard. Top utility is still SideKick, but there are dozens of clones of the Borland program now.

- I was glad to see one reviewer agree with me that WordPerfect is jolly slow — apparently it's something to do with the fact that the program is based on 8088/8086 technology.
- It seems as if most software here is sold mail order. But now software-only stores are opening up. I tried one called B Dalton Bookstores; some of the assistants seemed to know what they were talking about. Will this ever happen in the UK?
- Thrilled to see last week that my Washington DC hotel (The Willard) had computerised messaging and check-out systems in each room accessible via the TV remote control. It worked too. I checked out in nearly one second. The beauty of this: if you run through the lobby afterwards, how can they know you've just cleared the mini-bar?
- TORNADO NOTES is a sort of spreadsheet for the things you usually jot on a beer mat and then forget. Boggled? Review a'coming.
- Dongle disk copiers are here aplenty and doing well. But the 'look and feel' court-case recently has given Lotus the courage to sue absolutely everyone. Most satisfying to see that VisiCalc is now suing Locust for plagiarism.
- One Inmac-style catalogue given away here is called *The Supply Cabinet*. It's 816 pages long ...
- Renting computers by the hour, day etc is a doddle in the States. In New Orleans, friendly Dan Ellis will bring the computer and printer of your choice to your hotel, complete with interesting software, set it up and even provide disks and paper. Dan charges $85 a day; but speak to him nicely and you get to keep it a second day at no extra charge. Call him on (504) 455-5330.
- Scrupulous readers of my column will have revealed a preference for DISK OPTIMIZER. But I have just read a comparative review of these things — there are now around 15 which has got to prove something, right? — and DISK OPTIMIZER didn't do too well. And the review concluded that Norton Advanced crashes hard disks. Quoth Norton: ''We're working on it''. The one I bought was the Editor's Choice — Mace+ Utilities. So when I managed, yesterday, to wreck the FAT on Dan's XT, did it work at restoring the thing? It did, and within 45 seconds too. What a relief. Around $80 and worth every penny.
- Intriguing to note that some software here has a 'limited' lifetime guarantee. A free copy of my name and address to anyone who can guess that 'limited' means.
- Now I didn't know this (you probably did). If you've got a hard disk bigger than 10MB and you're using MS-DOS 3.x, you get smaller cluster sizes. Isn't this nice? I had been curious about Norton SI reporting 10K or so of clusters on my Apricot IBM clone; on my Xi20, running MS-DOS 2.11, there were only 2.4K or so. ◆

About the newsletter

Our aim is to provide a complete editorial and information service to users of Apricot computers.

Except in the normal course of day-to-day contact, sub sales and the like, we are not affiliated with or supported by Apricot plc or any of its dealers. Nor for that matter are we sponsored by or directly associated with any user group or any other such organisation, though our goals are similar to the declared aims of many such bodies.

The Publisher does not necessarily agree with the views expressed by contributors, and assumes no responsibility for errors of reproduction or interpretation in the subject matter of this newsletter or from any results arising therefrom.

APRICOT FILE is a publication of the TP Group — which until we get the name changed is the trading style of Permapoint Limited, registered in England no 1929195.

Postal address

TP GROUP
PO Box 509
London N1 1YL, England

Personal calls and deliveries

TP GROUP
Unit 4, 47-51 Wharfdale Road
London N1 9SE, England

Phone, E-mail, Telex

Telephone	**01-833 3501**
Telecom Gold	**81:JET767**
	[from overseas 10081:JET767]
Easylink	**19011170**
The Source	**BDE566**
Telex	**265871 MONREF G**
	quote reference 81:JET767

The Peoplealls and deliveries

Editor	**Dennis Jarrett**
Writer	**Garret Keogh**
Subscriptions	**Terrance Duggan**
DFR/Technical	**Roger Gann**

The Guarantee

The annual subscription to APRICOT FILE buys 12 monthly issues as follows:

Britain	£50.00
Rest of Europe	£60.00
Elsewhere (air mail)	£65.00

If having subscribed you come to feel that APRICOT FILE does not meet your requirements or that it does not match our claims, we will gladly cancel your commitment to the magazine and refund the unused portion of your subscription.

■■■ ISSN 0268-6066 ■■■

Act now!

An *aide-memoire* to support our feature in this issue: the first prosecution under the Data Protection Act has been successful. It does look a bit like using a truckload of ACT Siriuses to crush a fly — a Bedfordshire garage which kept customer records for marketing purposes was fined £500 for failing to register. It must be a case of *pour encourger les autres*; still, it's cheaper to sign up ...

Young man's fancy

"The Cabinet's most proficient computer user" is **Lord Young**. In the current issue of *Computerland Magazine*, the retail chain's house mag, a lengthy and somewhat sycophantic interview homes in on the man's computerish background — from the second Apple II in the country (running VisiCalc, ten years ago) to a current clutch of Compaq at home, Toshiba in the briefcase, and ICL PC in the office. And for word processing? "I use **WordPerfect**, which is by far the best". If Lord Young of Graffham uses WordPerfect, how can you possibly eschew it? We're checking the list of contributors to the Conservative Party right now ...

Compsoft appeals

It's losing contact with the Apricot-relevant days of DELTA and DMS, but **Compsoft** has long been one of our penny-share recommendations for APRICOT FILE readers looking to finance their resubscription by a flutter on the Stock Market. The latest reports are good: Compsoft has successfully completed a £1.5 million rights issue, the restructuring by new owners **Octagon Industries** appears firm, unprofitable European operations were shed late last year, the Which Computer Show saw several new products, and the shares — quoted on the USM — have been up to 100p. There's mileage in Compsoft yet ...

Another of our tips, **Brikat**, also continues to look good. The firm has just changed its name to **Pegasus Group** as a reflection of the new concentration on its traditional business — it too has shed loss-making diversifications and is majoring on the accounting packages it has always sold under the Pegasus name.

Apricot sheds some

The 1987 acquisition of LAN vendor **DMS** by **Apricot** has finally produced a few redundancies. Initially Apricot left all of the DMS operation in Wokingham, but at the start of the year downgraded its status to a maintenance centre and sales branch with the main activities relocating to Birmingham. Some **30** sales staff either didn't want to make the move or weren't invited to.

More inkjets

Inkjet printing has long had the air of a technology looking for a market. Some specialised applications like barcode printing have been around for a while; and as our recent review of the HP ThinkJet and Diconix 150 suggests, there's also a market for small, quiet printers in 'home office' uses (see APRICOT FILE Release 3.3). The Kodak subsidiary **Diconix** is however pushing for a much wider audience now — "we intend to be a full-line supplier to the computer printer market".

Hence the launch of the Diconix **300**, a go-faster brother of the 150 that claims the same standards of reliability and quietness (48dB, "less than the noise level in the average office") but zips along at a commendable 310cps in draft mode. The 300 comes in two models, both around 3in high and 9in deep; there's an 80-column unit at **£499** (width 14in, weight just 10lb) but the wide-carriage **300W** is expected to take the bulk of sales — all those executives wanted to bang out wide spreadsheets in near-silence at home or in the boardroom. The **300W** is 19in wide, weighs a shade under 13lb, and retails at **£599**.

These are high prices by comparison with dot matrix equivalents. But as well as the peacefulness, you do get two levels of NLQ (at 73cps or 49cps), a choice of two typefaces (the speeds quoted are for Elite — the option, Pica, slows down the printing a bit): dot-by-dot graphics: a single-sheet print option that means you don't have to remove any continuous pin-feed stationery already loaded: and the capability of printing on to a variety of different media — including OHP foils and labels as well as plain paper.

Distributors are **Systems Connections** on **0628 70912**, **Softsel** on **01-568 8866**, and **Norbain** on **0734 752201**.

Cheap colour

Elsewhere on the printer front, the new **Star LC-10** must be one of the cheapest seven-colour dot matrix printers in the UK now. For **£259** you get an 80-column printer with a draft speed of 144cps, a very decent NLQ mode (especially considering it's only a nine-pin print head) at 36cps, eight resident fonts, and low running costs — colour ribbons are £6, compared with the (optional) black-only ribbon from the colourless NL-10 at £4. This one also has a 'paper-parking' feature which means you can run single sheets without having to unload continuous stationery. There isn't an RS-232 version as yet, though. Star is on **01-840 1800**.

As a footnote to our front-page story, Artslink also says it will continue to keep in close touch with Apricot (which may or may not be a good thing) "to pass on information about software upgrades and products that might be of interest". And Artslink will be supplying upgrades to Communiqué version 2.0 for Micromail subscribers with the package.